

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-98367

(43)公開日 平成9年(1997)4月8日

(51)Int.Cl. <sup>a</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N	5/903		H 0 4 N	5/903
	5/91			5/91
	5/92			5/92
				B
				J
				H

審査請求 未請求 請求項の数17 O L (全 23 頁)

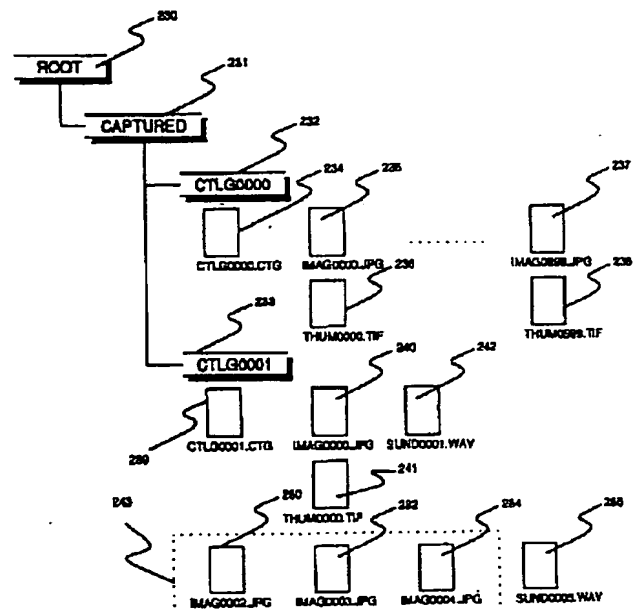
(21)出願番号	特願平7-256485	(71)出願人	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22)出願日	平成7年(1995)10月3日	(72)発明者	菅 章 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
		(72)発明者	山上 琢 東京都大田区下丸子3丁目30番2号 キヤノン株式会社内
		(74)代理人	弁理士 國分 孝悦

(54)【発明の名称】 信号処理装置

(57)【要約】

【課題】 デジタルカメラで撮像して記録された映像や音声のファイルの検索を容易にする。

【解決手段】 ROOTディレクトリ230の下にディレクトリ231を作り、その下に映像データ、音声データを保持するディレクトリ232、233を作る。ディレクトリ232の中には、この中の全データに関する属性情報を階層ヒープ構造で管理するマップファイル234、画像データファイル235、その縮小画像データファイル236との組合せ等がシリアル番号を付されて格納される。格納データが1000個に達するとシリアル番号がリセットされると共に、新たにディレクトリ233が作成され、その中にマップファイル239、画像ファイル240、その縮小画像ファイル241、3連字による画像ファイル280、282、284、その縮小画像ファイル281、283、285等が格納される。



【特許請求の範囲】

【請求項 1】 互いに種別が異なる第 1 種のデータ及び第 2 種のデータを取得し処理する信号処理手段と、上記信号処理手段で処理した第 1 種のデータ及び第 2 種のデータに対して、その種別に拘らず取得した順にシリアル番号を付加するシリアル番号付加手段とを備えた信号処理装置。

【請求項 2】 上記シリアル番号付加手段が付加したシリアル番号が所定値に達したときリセットするリセット手段を設けた請求項 1 記載の信号処理装置。

【請求項 3】 上記所定値は、上記シリアル番号を表示手段で表示する際の表示可能な値であることを特徴とする請求項 2 記載の信号処理装置。

【請求項 4】 上記信号処理手段が処理したデータにデータ名を付加するデータ名付加手段を設けた請求項 1 記載の信号処理装置。

【請求項 5】 上記信号処理手段は、取得した第 1 種のデータ及び第 2 種のデータから各々のファイルを作成する処理を行うと共に、上記各ファイルにファイル名を付加するファイル名付加手段を設けたことを特徴とする請求項 1 記載の信号処理装置。

【請求項 6】 上記信号処理手段が取得したデータに関する属性情報を保持する属性情報保持手段を設けた請求項 1 記載の信号処理装置。

【請求項 7】 上記信号処理手段が処理したデータを上記シリアル番号と共に記憶する記憶手段を設け、上記属性保持手段は、上記シリアル番号とこのシリアル番号に対応するデータの格納場所及び他の属性情報を取得するための情報を格納するデータ取得履歴テーブルを有することを特徴とする請求項 6 記載の信号処理装置。

【請求項 8】 上記シリアル番号とデータ消去が指定されたとき、上記記憶手段における上記指定されたシリアル番号のデータを消去すると共に、上記データ取得履歴テーブルの上記消去したデータに関する情報を消去するようにした請求項 7 記載の信号処理装置。

【請求項 9】 指定されたシリアル番号のデータに対応する属性情報を上記データ取得履歴テーブルのデータに基づいて取得し表示することを特徴とする請求項 7 記載の信号処理装置。

【請求項 10】 複数の可変長のデータを格納可能な第 1 の領域と、格納するデータ数と各データの取得情報とを格納可能な第 2 の領域と、上記第 2 の領域の位置を格納する第 3 の領域より構成されるヒープ構造を階層的に用いてデータを格納する階層ヒープ構造により、格納した上記各データの格納場所及び各データに対応した属性情報を格納する属性情報保持手段を設けたことを特徴とする信号処理装置。

【請求項 11】 上記属性情報保持手段が、上記第 2 の領域の取得情報中に直接データを格納する格納手段を有することを特徴とする請求項 10 記載の信号処理装置。

【請求項 12】 上記属性情報保持手段が、ヒープ中に上記データを格納する外部ファイル識別情報を保持することによって間接的に上記データを格納する格納手段を有することを特徴とする請求項 10 記載の信号処理装置。

【請求項 13】 上記属性情報保持手段が、第 2 階層以上のヒープ中にデータを格納する際に第 1 階層のヒープ中のデータの識別情報を上記第 2 の領域中に保持することによって深い階層のヒープ中のデータと第 1 階層のデータとを連結させてデータを格納する格納手段を有することを特徴とする請求項 10 記載の信号処理装置。

【請求項 14】 複数の属性情報をグループ化して 1 つのヒープ構造中にそのグループ化された属性情報をまとめて 1 段深い階層のヒープ構造中に保持する保持手段を有し、上記グループ化された属性情報の 1 つのメンバーとして、さらに複数の属性情報をグループ化したサブグループを有することができ、サブグループ情報はさらに 1 段深い階層のヒープ構造中に保持する保持手段を有し、同様に属性情報を N 階層に階層化して保持する保持手段を有することを特徴とする請求項 10 記載の信号処理装置。

【請求項 15】 属性情報を保持する際に高速に読み込む必要があるもしくは読み込む頻度が高い程浅い階層に配置し、高速に読み込む必要がないもしくは読み込む頻度が低い程深い階層に配置する配置手段を有することを特徴とする請求項 14 記載の信号処理装置。

【請求項 16】 読み込む頻度もしくは読み込み速度への要求の高さに応じて属性情報をグループ化し、この属性情報のグループを保持する際に高速に読み込む必要があるもしくは読み込む頻度が高い程浅い階層に配置し、高速に読み込む必要がないもしくは読み込む頻度が低い程深い階層に配置する配置手段を有することを特徴とする請求項 14 記載の信号処理装置。

【請求項 17】 複数の画像で一つの撮影単位となる複数の画像を各画像の属性情報を含むサブグループとし、さらに上記複数のサブグループより構成される撮影単位を上記撮影単位全体の属性情報を含むグループとして保持する保持手段を有することを特徴とする請求項 14 記載の信号処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は映像データ及び音声データをキャプチャーして処理する信号処理装置に関し、特に映像、音声をデジタル記録するデジタルカメラに適用して好適な信号処理装置に関するものある。

【0002】

【従来の技術】 近年、映像や音声をキャプチャーし、メモリーカード等の記録媒体に記録するデジタルカメラが開

ルとして記録媒体に記録する。また、個々のファイルにはそのキャプチャーデータに関連する属性情報、例えば撮影年月日、撮影モード、撮影条件等も記録される。これらの画像ファイルや音声ファイルを管理する方法として、各キャプチャーデータの属性情報を1つのマップファイルに記録する方法がすでに提案されている。

【0003】

【発明が解決しようとする課題】デジタルカメラの特徴として、画像や音声データをキャプチャーするだけでなく、様々な属性情報をキャプチャーデータに関連づけて管理し、画像を表示する際に、画像とともに表示したり、属性情報を検索キーとして所望の画像を検索したりする機能が期待される。しかしながら、画像ファイルや音声ファイルの属性情報は固定長のデータだけでは表現できず、可変長のデータを効率よく管理しなければならない。例えば、画像を説明するテキスト情報を画像に属性情報としてつける際も可変長データとなる。また、検索に用いるようなデータは極力高速に取り出すことが可能でなければならない。また、デジタルカメラを低価格にするために少ないメモリで可変長のデータを管理しなければならない。また、画像をキャプチャーする場合、連写撮影時やパノラマ撮影時には複数の画像を一つの撮影単位として撮影者が認識できるようにマップファイル中で管理する要求があった。

【0004】また、デジタルカメラにおいては、静止画や音声など様々なデータをキャプチャーする。そしてそれらのキャプチャーデータのうち所望のデータを指定して、再生したり消去したりする機能が望まれるが、小型・軽量・低価格を実現するために簡便な表示装置しか備えない場合が多い。その場合でも所望の画像や音声を選択する機能が必要である。また、様々なキャプチャーデータ、属性情報をマップファイルで管理する場合、マップファイルに基づいて簡単な方法で所望のキャプチャーデータを検索する必要がある。また、デジタルカメラにおいて、記録媒体として大容量のものをを用いた場合、従来の銀塩カメラと比較して非常に多くの枚数の画像データを撮影することができる。その場合、表示装置の数字の表示桁数が限られていても、表示装置の数字の表示桁数以上の画像データをキャプチャーできる必要がある。

【0005】そこで、本発明の目的は、多くの映像データや音声データから所望のデータを簡単に検索することのできる信号処理装置を得ることにある。

【0006】

【課題を解決するための手段】請求項1の発明においては、互いに種別が異なる第1種のデータ及び第2種のデータを取得し処理する信号処理手段と、上記信号処理手段で処理した第1種のデータ及び第2種のデータに対して、その種別に拘らず取得した順にシリアル番号を付加するシリアル番号付加手段とを設けている。

【0007】請求項10の発明においては、複数の可変

長のデータを格納可能な第1の領域と、格納するデータ数と各データの取得情報とを格納可能な第2の領域と、上記第2の領域の位置を格納する第3の領域より構成されるヒープ構造を階層的に用いてデータを格納する階層ヒープ構造により、格納した上記各データの格納場所及び各データに対応した属性情報を格納する属性情報保持手段を設けている。

【0008】

【作用】本発明によれば、シリアル番号付加手段は、信号処理手段で処理したデータに、そのデータが映像・音声に拘らずデータを取得した順番に番号を付けていく。従って、このシリアル番号が付されたデータを記録媒体に記録したとき、シリアル番号に基づいてデータを容易に検索することができる。

【0009】

【発明の実施の形態】第2図はデジタルカメラの構成を示すブロック図である。図2において、101はレンズ、102はレンズ101によって結像された光学像を電気信号に変換する撮像素子、103は撮像素子102の出力をAD変換するAD変換回路、104はAD変換回路103の出力を一時的に記憶するメモリ、113はメモリ104に一時記憶されたデータから色信号処理等の信号処理をする信号処理部。105は信号処理部113より得られた画像データを圧縮符号化する圧縮符号化部である。108はフラッシュメモリ等の記録媒体であり、106は記録媒体108に圧縮画像データを記録する際の記録インターフェースである。107はデジタルカメラの動作を制御するCPUであり、114はCPU107が用いるメインメモリ、109はデジタルカメラの設定や撮影動作を利用者が行うための操作部である。110はマイクロホン等の音声入力部であり、111は音声入力部110の信号をAD変換する音声AD変換回路である。112は音声データを符号化する音声符号化部であり、符号化された音声データは記録インターフェース106を介して記録媒体108に記録される。

【0010】図3はデジタルカメラの背面に配置した操作部109の例を説明する図である。図3において、120はデジタルカメラの背面、121は操作系表示部である。122はキャプチャーした音声または映像を識別するシリアル番号表示である。操作系表示部121はデジタルカメラを小型・軽量・低価格とするために、簡便なものが用いられることが多く、シリアル番号表示122の表示桁数は3桁程度の場合が多い。また、利用者が数字を記憶する場合、3桁を超えると記憶が困難になることからシリアル番号の表示桁数は3桁に限定するのが良い。130は表示されているシリアル番号のデータが映像である場合はP、音声である場合はSと表示されるキャプチャーデータ種別表示である。129はモード設定スイッチであり、OFF（電源オフ）、Rec（記録モード）、Erase（消去モード）の3モードを選択できる。

123及び124は記録データのシリアル番号の選択をするためのそれぞれダウンボタン、アップボタンである。125は消去ボタンであり、モード設定スイッチ129が消去モードにある場合に、この消去ボタン125を押すことによって表示されているシリアル番号の音声または画像データを消去する。126は撮影のトリガとなるリリーススイッチである。127はマイクロホン、128は音声録音スイッチである。

【0011】図4は本発明において後述するマップファイルに可変長のデータを格納するための基本構造として用いるヒープ構造300を説明する図である。図4において、309～312はN個の可変長のデータレコードを示している。データレコードは一つのデータのまとまりを示す単位であり、単にレコードと呼ぶこともある。309はデータレコード1、310はデータレコード2、311はデータレコードN、312はデータレコード7となっている。301はヒープに含まれるレコード数と各データレコードのレコード型と、オフセットと、レコード長を示すデータレコード取得情報をテーブルにしたオフセットテーブルである。302は一つのヒープに含まれるデータレコード数である。303はデータレコード1のデータレコード取得情報であり、データレコード1の型を示すレコード型フィールド304とヒープの先頭からデータレコード1の先頭までのオフセット量を示すレコードオフセットフィールド305とデータレコード1の長さを示すレコード長フィールド306とから構成される。これらのデータレコード取得情報をもとにしてデータレコードを読み取ることが可能になる。同様に307はデータレコード2のデータレコード取得情報、308はデータレコードNのデータレコード取得情報である。350はオフセットテーブルの位置を取得するためのオフセットテーブルオフセットである。このようにヒープ構造300はN個の可変長データレコードとオフセットテーブル301とオフセットテーブルオフセット350とから構成される。

【0012】図5はレコード型フィールド304の一般形を示した図である。図5において、335はデータレコードの格納方法を識別するためのビットフィールドを示しており、type.stgと略記する。336はデータレコードの概略の種別を示すビットフィールドであり、type.dataTypeと略記する。このビットフィールドによってデータレコードがASCIIであるかWORDであるか等の大まかなデータ形式の分類を示す。337はデータレコードの詳細な種別を示すIDコードビットフィールドでtype.IDCodeと略記する。

【0013】図6は図4のヒープ構造300を拡張した階層ヒープ構造を示した図である。本発明では図6の階層ヒープ構造によってマップファイルを構成している。351は階層ヒープ構造の一例を示している。353は第1階層の一つのデータレコードであるデータレコード

1であるがデータレコード1(353)はさらにヒープ構造357で第2階層のヒープとして構成されている。このようにヒープ構造を階層化していくことからこのような構造を階層ヒープ構造と呼ぶ。さらにこの階層ヒープ構造においては全部で4種類のデータレコード格納方法を実現しており、この格納方法の種別を図5で説明したtype.stgビットフィールド335によって識別する。次にこれら4つのデータレコード格納方法について説明する。

【0014】第1のデータレコード格納方法は、ヒープ中にデータを格納する方法であり、図4で説明した方法である。例としてデータレコード3(354)はこの方法でデータを格納している。データレコード3(354)のデータレコード取得情報363中のレコード型フィールド316のtype.stgビットフィールドに定数klnHeapSpaceを指定することでヒープ中にデータが格納されていることが識別される。レコードオフセット317にはヒープの先頭からデータレコード3(354)の先頭までのオフセット量が格納されている。

【0015】第2のデータレコード格納方法は、オフセットテーブル中のレコードオフセットとレコード長のフィールドとを用いてこの範囲に納まるデータをオフセットテーブルのデータレコード取得情報中に格納する方法である。図6の階層ヒープ351中4番目のデータレコードがその例である。オフセットテーブル355中データレコード4のデータレコード取得情報364のレコード型フィールド318中のtype.stgビットフィールドがklnRecordEntryという定数である場合、レコードオフセットとレコード長を格納するべき領域319にデータレコードを直接格納する。このように小さなデータレコードは直接オフセットテーブル中に格納することで、データ格納スペースが節約できるのみならずデータレコードのアクセス速度が向上する。

【0016】第3のデータレコード格納方法は、第2階層以上の深い階層のヒープ中のメンバーとして第1階層のレコードのインデックスを持つことによって第1階層のレコードと連結させる方法である。第2階層のデータレコード2がこの格納方法で第1階層のデータレコード5(356)と連結されている。第2階層のヒープ357のオフセットテーブル326中のデータレコード取得情報329中のレコード型フィールド333のtype.stgビットフィールドに定数klnConnectedFirstLevelRecordを指定することで、このデータレコード格納方法が識別される。レコードオフセットを格納する領域334に連結された第1階層のデータレコードのインデックスを格納することで、どのデータレコードと連結されるかを識別する。この例では第1階層のデータレコード5と連結されている。

【0017】このデータ格納方法によれば、大きな子データレコードによる子階層ヒープの増大を防ぐことがで

きる。また、この方法によれば、子階層には大きなデータを直接持たないので第2階層ヒープ357を全てメモリ中に持つことができる。連結されたデータレコード5(356)が必要になったときのみデータレコード5(356)を読み込むようにすることで、メインメモリ114の効率的な使用が可能になる。さらにこのようにデータを格納することで1つのファイル中に階層的にデータを格納する際に階層構造に従って連続的にデータを配置する必要はないので、階層構造の変更の際にファイル内でのデータの移動量を最小化することができる。例えば第1階層のあるデータレコードAを第2階層のデータレコードBのメンバーから第3階層のデータレコードCのメンバーに移動する際に連結するデータのインデックスを書き換えるだけでデータの移動が完了する。例えばあるデータレコードの属性情報として、アイコン画像を、マップファイル上に格納する場合、この方法で格納することにより、親のデータレコードを移動削除する際にデータ量の大きいアイコン画像を移動しなくてよいという利点がある。

【0018】第4のデータ格納方法は実際のデータレコードを外部ファイルに格納する方法である。第1階層のデータレコード2(352)がその例である。データレコード2(352)のデータレコード取得情報362のレコード型フィールド314のtype.stgフィールドに定数klnExternalFileが指定されていると、実際のデータレコードが外部ファイルに格納されていることが示される。レコードオフセット315が示すデータレコード2には外部ファイル名が格納されており、この情報により実際のデータを格納した外部ファイル320中のデータを取得することができる。このデータ格納方法によれば、大きなデータを外部ファイルに格納し、ヒープ中のデータレコードと連結することができ、マップファイルの容量を小さくできるのみならず、様々なフォーマットの外部ファイルをマップファイル中のデータレコードとして持つことができる。

【0019】図7は階層ヒープからのデータレコードのアクセス方法のフローチャートである。ステップs101においてヒープ末端のオフセットテーブルオフセットよりオフセットテーブルの開始位置を取得する。ステップs102においてオフセットテーブルのレコード数フィールドよりレコード数を取得する。次にs103において全レコードを取得すれば終了し、全レコードを取得するまで以下のステップを繰り返す。ただしオフセットテーブルを読んで必要なデータレコードだけを読むようにしても良い。ステップs104において次レコードのレコード型をデータレコード取得情報のレコード型フィールドから取得する。ステップs105においてtype.stg=klnHeapSpaceの場合ステップs106でデータレコード取得情報のレコードオフセットフィールドとレコード長フィールドを読み取り、その情報をもとにステップ

s107においてデータレコードをヒープより取得する。ステップs112でtype.stg=klnRecordEntryの場合ステップs113でデータレコードをオフセットテーブル中より取得する。

【0020】ステップs108においてtype.stg=klnConnectedFirstLevelRecordであった場合、ステップs109においてレコードオフセットフィールドより連結データレコードのインデックスを取得する。次にステップs110において連結データレコードのインデックスに対応した第1階層のデータレコード取得情報を第1階層のオフセットテーブルより取得する。次にステップs111においてデータレコードを第1階層ヒープより取得する。ステップs114においてtype.stg=klnExternalFileであった場合は、ステップs115で外部連結ファイル名を格納するデータレコードのレコードオフセット、レコード長を取得する。次にステップs116において外部ファイル名をヒープより取得する。次にステップs117においてデータレコードを外部ファイルより取得する。

【0021】図1はデジタルカメラの記録媒体108にキャプチャーデータを記録していく際のディレクトリ及びファイル構成を説明する図である。この実施の形態では記録媒体108を例えばFATファイルシステムによってファイル管理している。ROOTディレクトリ230の下にディレクトリCAPTURED(231)を作り、その下に1000個までの単位で映像または音声データ等のキャプチャーデータを保持するディレクトリを作る。図1におけるディレクトリCTLG0000(232)、ディレクトリCTLG0001(233)がキャプチャーデータ保持ディレクトリの例である。ディレクトリCTLG0000(232)中には、このディレクトリCTLG0000(232)中の全キャプチャーデータの属性情報を管理するマップファイルCTLG0000.CTG(234)、画像データファイルIMAG0000.JPG(235)とそのサムネール画像データファイルであるTHUM0000.TIF(236)との組み合わせから画像データファイルIMAG0999.JPG(237)とそのサムネール画像データファイルであるTHUM0999.TIF(238)との組み合わせにいたる1000個までのキャプチャーデータが格納される。ここでサムネール画像データとはキャプチャーした画像データをパーソナルコンピュータ等に読み込む際に、全キャプチャーデータの一覧表示等に用いる縮小画像である。

【0022】ディレクトリCTLG0000(232)に1000個のキャプチャーデータが格納されるとディレクトリCTLG0001(233)が自動的に作成され、ディレクトリCTLG0001(233)中の全てのキャプチャーデータを管理するマップファイルCTLG0001.CTG(239)が自動的に作成される。次に最初にキャプチャーされた画像データファイルであるIMAG0000.JPG(240)が作成され、そのサムネール画像ファイルであるTHUM0000.TIF(24

1)、次にキャプチャーされた音声データファイルであるSUND0001.WAV(242)、次にキャプチャーされた画像データファイルであるIMAG0002.JPG(280)とそのサムネイル画像ファイルTHUM0002.TIF(281)、次にキャプチャーされた画像データファイルであるIMAG0003.JPG(282)とそのサムネイル画像ファイルTHUM0003.TIF(283)、次にキャプチャーされた画像ファイルであるIMAG0004.JPG(284)とそのサムネイル画像ファイルであるTHUM0004.TIF(285)、次にキャプチャーされた音声ファイルであるSUND0005.WAV(286)が格納されている。ここで画像ファイルIMAG0002.JPG(280)、IMAG0003.JPG(282)、IMAG0004.JPG(284)の3枚の画像は1回の連写撮影で撮られた1組の連写画像243であるとする。

【0023】図8はマップファイルCTLG0001.CTG(239)中に格納されるデータ取得履歴テーブルの説明図である。Nobjは一つのキャプチャーデータ格納ディレクトリ中、キャプチャー順に付与されるシリアル番号であり、オブジェクトシリアル番号と称する。マップファイル中では一つのデータのまとまりをオブジェクトと称してマップファイルを構成する第1階層のヒープのデータレコードとしてデータを管理する。例えば連写画像243はこれを構成する3枚の画像をサブオブジェクトとし、それらのサブオブジェクトより構成された一つのオブジェクトとして管理する。オブジェクトはマップファイルの第1階層のヒープ中のデータレコードとして管理され、object indexはマップファイルCTLG0001.CTG(239)中の第1階層のヒープ中におけるデータレコードのインデックスを示す。subObject indexはオブジェクトを構成するサブオブジェクト、例えば連写画像中の各画像のインデックスを示す。図8の例ではNobj=0が画像ファイルIMAG0000.JPG、Nobj=1が音声ファイルSUND0001.WAV、Nobj=2が画像ファイルIMAG0002.JPG、Nobj=3が画像ファイルIMAG0003.JPG、Nobj=4が画像ファイルIMAG0004.JPG、Nobj=5が音声ファイルSUND0005.WAVに付与されている。

【0024】ここでファイル名はオブジェクトシリアルナンバーを含めるように構成する。例えばNobj=0の画像ファイルのファイル名IMAG0000.JPGは、画像であることを示すIMAGとオブジェクトシリアルナンバーNobjを4桁で示した0000と画像のデータ形式を示す拡張子JPGから構成される。2番目にキャプチャーされた音声データファイルSUND0001.WAVは音声であることを示すSUNDとオブジェクトシリアルナンバーNobjを4桁で示した0001と音声のデータ形式を示す拡張子WAVから構成される。

【0025】図9はマップファイルCTLG0001.CTG(239)中に格納されるデータレコードを説明する図である。マップファイルCTLG0001.CTG(239)は階層ヒープ構造によって構成されている。ここで第1階層ヒープに格納されるデータレコードをオブジェクトと称してい

る。図中202はobject index=1のマップ管理オブジェクトであり、マップファイルの管理情報が納められる。203は図に示したデータ取得履歴テーブルであり、マップ管理オブジェクトの中の1つのデータレコードとして格納される。データ取得履歴テーブルのようにオブジェクトがデータレコードとして含む属性情報をプロパティと称することにする。オブジェクトの中にはそのオブジェクトの様々な属性を示す各種のプロパティが格納されるが、図9の説明においては代表的なプロパティのみにについて説明する。

【0026】204はobject index=2のイメージオブジェクトであり、単写であるのでsubObject index=1の1個の画像データであるサブオブジェクト205が含まれる。206はobject index=3のサウンドオブジェクトである。207はobject index=4のイメージオブジェクトであり、イメージオブジェクト207は3枚の画像の連写なのでsubObject index=1の画像データであるサブオブジェクト208とsubObject index=2のサブオブジェクト209とsubObject index=3のサブオブジェクト210とから構成される。211はobject index=5のサウンドオブジェクトである。212はマップファイルCTLG0001.CTG(239)の第1階層のデータレコードの取得情報を格納するオフセットテーブルであり、213はオフセットテーブル212のマップファイルCTLG0001.CTG(239)内における位置を得るためのオフセットテーブルオフセットである。オフセットテーブルオフセット213にはオフセットテーブル212のマップファイルCTLG0001.CTG(239)の先頭からのオフセットアドレスであるa5が格納されている。

【0027】図10はオフセットテーブル212の構成を示す図である。図10において、214はレコード数であり、この例ではレコード数=5となる。215はobject index=1のデータレコードのデータレコード取得情報であり、レコード型フィールド220のtype、IDCodeビットフィールドは定数mapManagementObjectとなりマップ管理オブジェクトであることが示される。このデータレコードは図9のごとくファイルの先頭に位置しているので、レコードオフセットフィールド221は0となっている。216はobject index=2のデータレコードのデータレコード取得情報であり、レコード型フィールド222のtype、IDCodeビットフィールドは定数imageObjectとなっていることが識別される。

【0028】レコードオフセット223は図9のごとくa1となっている。217はobject index=3のデータレコードのデータレコード取得情報であり、レコード型フィールド224のtype、IDCodeビットフィールドは定数soundObjectとなっていることが識別される。レコードオフセットは図9のごとくa2となっている。218はobject index=4のデータレコードの

データレコード取得情報であり、レコード型フィールド 226 の type. IDCode ビットフィールドは定数 imageObject となってイメージオブジェクトであることが識別される。レコードオフセット 227 は図 9 のごとく a3 となっている。219 は object index=5 のデータレコードのデータレコード取得情報であり、レコード型フィールド 228 の type. IDCode ビットフィールドは定数 soundObject となってサウンドオブジェクトであることが識別される。レコードオフセットフィールドは図 9 のごとく a4 となっている。

【0029】図 11 はマップファイル CTLG0001.CTG (239) 中のイメージオブジェクト 207 の構成を説明する図である。イメージオブジェクト 207 は第 2 階層のヒープによって構成される。第 2 階層のヒープはサブオブジェクト 208 とサブオブジェクト 209 とサブオブジェクト 210、イメージオブジェクト 207 の説明テキスト情報を格納するディスクリプションプロパティ 292、オフセットテーブル 250、オフセットテーブル オフセット 251 から構成される。252 はオフセットテーブル 250 のレコード数フィールドでこの例では 5 となる。253 は第 1 のデータレコードのデータレコード取得情報であり、レコード型フィールド 257 の type. stg ビットフィールドが定数 kInRecordEntry、であることからレコードオフセットフィールド及びレコード長フィールド 258 に直接データが格納されている。type. IDCode ビットフィールドが定数 thumbnailSubObjectID であることからデータは連写画像のうちどの画像（サブオブジェクト）のサムネールを連写画像を代表するサムネールとして用いられるかを指定する。この例ではレコードオフセットフィールド 258 に thumbnailSubObjectID=1 が格納されているので、連写画像中 1 枚目の画像のサムネールを連写画像のサムネールとして用いることが示される。

【0030】254 は第 2 のデータレコードのデータレコード取得情報であり、レコード型フィールドの type. stg ビットフィールドが定数 kInHeapSpace であることからヒープにデータレコードが格納されている。レコードオフセットフィールド 260 はヒープ中のサブオブジェクト 208 の先頭アドレスのヒープの先頭からのオフセットを示す。同様に 255、256 は第 3、第 4 のデータレコード、すなわちサブオブジェクト 209 とサブオブジェクト 210 の取得情報を示す。293 は第 5 のデータレコードのデータレコード取得情報であり、レコード型フィールド 295 の type. stg ビットフィールドが定数 kInHeapSpace であることからヒープにデータレコードが格納されている。また、type. IDCode ビットフィールドが定数 description であることからこのデータレコードは画像のディスクリプションすなわち説明情報であることがわかる。レコードオフセット 296 はヒープ上のディスクリプションプロパティ 292 のヒープの先頭から

のオフセットを示す。

【0031】このように連写のように複数の画像を 1 つの撮影単位として管理することが可能になり、デジタルカメラでキャプチャーした画像をパーソナルコンピュータの画面上にサムネールを一覧表示する際に連写した複数の画像を 1 枚のサムネールで代表することができ、画像の検索がしやすくなる。また、ディスクリプション等の付帯情報も連写の 1 撮影単位に対してまとめて付与することができ、データ容量を節約することができる。

【0032】この実施の形態では連写を 1 つのイメージオブジェクトとして表現したが、1 枚の大きな画像を複数の部分画像に分割して撮影するパノラマ撮影等も複数の画像を 1 つのイメージオブジェクトとして表現することによって同様の効果がある。

【0033】図 12 はサブオブジェクト 208 のデータレコードの内部構成を示した図である。サブオブジェクト 208 も 3 階層目のヒープ構造によって構成される。図 12 の例ではヒープ上に積まれたデータが 3 個、オフセットテーブル中に埋め込まれたデータが 2 個で計 5 個のデータレコードがある。そこでオフセットテーブル 264 中のレコード数フィールド 265 は 5 となる。第 1 のデータレコード取得情報 267 はレコード型フィールド 272 の情報よりオフセットテーブル中に subObject index を格納し値は 1 である。すなわちこの情報からイメージオブジェクト中の 1 枚目の画像であることがわかる。ただしイメージオブジェクトが単写の場合は subObject index=1 の画像しか存在しない。第 2 のデータレコード取得情報 268 はレコード型フィールドの情報よりレコードオフセットフィールド 275 にオブジェクトシリアルナンバー Nobj を格納し値は 2 である。第 3 のデータレコード取得情報 269 はレコード型フィールド 276 の情報より外部ファイルに imageData すなわち画像データを持つことを示している。

【0034】レコードオフセット 277 が示すヒープ上のデータレコード 262 には画像データのファイル名が格納されており、画像ファイル IMAG0002.JPG (280) を指し示す。第 4 のデータレコード取得情報 270 はレコード型フィールド 278 よりレコードオフセット 279 が示すヒープ上のデータレコード 263 に shootingRecord すなわち撮影記録情報を格納していることを示す。撮影記録情報はここでは詳細に説明しないが、さらにヒープ構造によって撮影時のシャッタースピードや絞り値等に関する情報を格納する。第 5 のデータレコード取得情報 271 はレコード型フィールド 290 の情報よりレコードオフセット 291 が示すヒープ上のデータレコード 261 に thumbnail すなわちサムネール画像ファイルのファイル名が格納されており、そのファイル名はサムネール画像ファイル THUM0002.TIF (281) を指し示す。

【0035】図 13 はイメージオブジェクトを階層ヒ-

ブ構造のマップファイル中に格納する際に各プロパティの階層構造を示した図である。図中400はイメージオブジェクト、401はサムネールオブジェクトID、402はディスクリプション、403はサブオブジェクト、404はサブオブジェクトインデックス、405はオブジェクトシリアルナンバー、406はイメージデータ、407はサムネール、408は撮影記録情報、409はシャッタースピード値、410は絞り値である。これらのプロパティのうちディスクリプション402には検索のためのキーワードを格納可能であるが、浅い階層に格納することで高速に取り出すことができる。また、シャッタースピード値409等は表示には用いるが検索には用いないため深い階層に格納し、必要のないときは読み飛ばすようにする。デジタルカメラを安価にするためにはCPU107が用いるメインメモリ114は最小限にする必要がある。マップファイルからデータを読む場合も必要なデータだけをメインメモリ114に読み込み不要なデータは読み飛ばすようにすることが必要である。その際データが種別によって階層ヒープ中のデータレコード単位でまとまっているので、読み込む必要のありなしを全てのデータに対してチェックする必要性がなくなり、高速に判断ができるようになる。例えば検索のときに撮影記録情報408を用いないのであれば、撮影記録情報408はまとめて読み飛ばすことができ、撮影記録情報408がどのようなデータで構成されているかなどのチェックはまったくしなくて良い。

【0036】このように高い読み込み速度が要求される属性情報は浅い階層に配置し、高い読み込み速度が要求されない属性情報は深い階層に配置することで、マップファイル中の属性情報を用いた検索等の処理を高速化できる。さらに、高い読み込み速度が要求される属性情報はグループ化して浅い階層に配置し、高い読み込み速度が要求されない属性情報はグループ化して深い階層に配置することで、読み込みの必要の有りの判断を属性情報毎に行わずに済むため、マップファイル中の属性情報を用いた検索等の処理をさらに高速化できる。また、読み込む頻度が高い属性情報は浅い階層に配置し、読み込む頻度が低い属性情報は深い階層に配置することで、マップファイル中のデータ表示等の処理を高速化できる。さらに、読み込む頻度が高い属性情報はグループ化して浅い階層に配置し、読み込み頻度が低い属性情報はグループ化して深い階層に配置することで、読み込みの必要の有りの判断を属性情報毎に行わずに済むため、マップファイル中のデータ表示等の処理をさらに高速化できる。

【0037】図14は撮影時すなわちモード設定スイッチ129がRecモードの際のファイル管理のシークエンスを示す図である。図14においてステップs1において電源オンするとステップs2においてカレントディレクトリを決定する。次にステップs3でイベント待ちと

なり撮影者の操作を待つ。ステップs4でイベントがあればステップs5でイベント解析を行い、ステップs6で撮影か録音動作でなければステップs7の判断ステップに進み、電源オフ動作であれば電源をオフして終了する。電源オフ動作でなければステップs24で他の処理を行ってイベント待ちを行う。ステップs6で撮影か録音動作であった場合、ステップs9においてオブジェクトシリアルナンバーNobjが予め決められた最大値Nobjmaxを超えた場合ステップs10において新規ディレクトリを作成し、そのディレクトリをカレントディレクトリにし、カレントディレクトリに新規マップファイルを作成しNobjを0に初期化する。

【0038】例えば図1においてディレクトリCTLG0000(232)においてマップファイルCTLG0000.CTG中のオブジェクトシリアルナンバーNobjが999を超えると、ディレクトリCTLG0001(233)を作成し、そのディレクトリをカレントディレクトリとし、新規なマップファイルCTLG0001.CTG(239)を作成しオブジェクトシリアルナンバーNobjを0にリセットする。次にステップs11においてデータ取得履歴テーブル203を更新し、ステップs12で撮影動作であった場合はステップs13でNobjを含むファイル名で画像ファイルを作成し、ステップs14でNobjを含むファイル名でサムネールファイルを作成し、ステップs22でマップファイル中にイメージオブジェクトを記録した後、ステップs8でオブジェクトシリアルナンバーNobjを1インクリメントし、イベント待ちに戻る。ステップs12で撮影動作でない場合は録音動作であるのでステップs15でNobjを含むファイル名で音声ファイルを作成し、ステップs23でマップファイル中にサウンドオブジェクトを記録した後、ステップs8でオブジェクトシリアルナンバーNobjを1インクリメントし、イベント待ちに戻る。

【0039】図15は図14のステップs2のカレントディレクトリ決定の際のより詳細なフローチャートを示した図である。ステップs16においてディレクトリ名CAPTUREDの下に頭4文字がCTLGで始まり4桁の数字が続くディレクトリを検索し、ステップs17で該当ディレクトリがあればステップs18において下4桁の数字が最大のディレクトリを検索しカレントディレクトリとする。次にステップs19においてカレントディレクトリのマップファイルより次にキャプチャーするデータのオブジェクトシリアルナンバーを取得する。ステップs17で該当ディレクトリがなかった場合ステップs20でディレクトリCAPTUREDの下にディレクトリCTLG0000を作成しカレントディレクトリとする。次にステップs21でマップファイルCTLG0000.CTGを作成しオブジェクトシリアルナンバーを0に初期化する。オブジェクトシリアルナンバーはシリアル番号表示122に表示される。

【0040】このようにオブジェクトシリアルナンバーNobjがシリアル番号表示122で表示可能な桁数を超え



た場合は、新規なデータの格納単位を自動的に生成するので、大容量の記録媒体を使用した場合、シリアル番号表示122の表示桁数を上回る数のデータをキャプチャーすることが可能になる。キャプチャーしたデータを消去する際は、モード設定スイッチ129をEraseにすることによってダウンボタン123、アップボタン124の操作によってオブジェクトシリアルナンバーを増減し、消去したいキャプチャーデータのオブジェクトシリアルナンバーを選択する。選択中のオブジェクトシリアルナンバーはシリアル番号表示122に表示される。そのオブジェクトシリアル番号のキャプチャーデータのオブジェクトインデックスをデータ取得履歴テーブル203から取得しマップファイルから該オブジェクトシリアル番号のキャプチャーデータの種別を得てキャプチャーデータ種別表示130に画像ならP、音声ならSと表示する。このキャプチャーデータ種別表示130によって、何番目に取ったデータかというだけでなく、画像か音声かのデータの種別が分かるため、キャプチャー順とデータ種別の2つの情報より消去すべきデータかどうかの判断ができるため、誤消去の可能性が低くなる。消去ボタン125が押されるとマップファイルからそのデータのファイル名を得て該当ファイルを消去する。

【0041】図16は消去対象キャプチャーデータのオブジェクトシリアルナンバーを得てから後のデータ消去のフローチャートを示した図である。ステップs24においてデータ取得履歴テーブルより消去対象キャプチャーデータのオブジェクトインデックスおよびサブオブジェクトインデックスを取得する。ステップs25においてマップファイルよりオブジェクトインデックス及びサブオブジェクトインデックスに対応するデータのファイル名を取得する。ステップs26において該当オブジェクトが画像であればステップs27において該当する画像ファイルを消去しマップファイル上からサブオブジェクトのデータレコードを消去する。その結果ステップs28でサブオブジェクト数が0となったらマップファイル上からイメージオブジェクトデータレコードを消去し、ステップs30においてデータ取得履歴テーブルを更新し終了する。ステップs26において該当オブジェクトが画像でなく音声であった場合はステップs31において該当する音声ファイルを消去しマップファイル上からサウンドオブジェクトデータレコードを消去し、ステップs30でデータ取得履歴テーブル203から消去データに関する情報を消去することによって更新して消去を終了する。

【0042】このようにマップファイルに多数のデータが格納されていても、簡単な表示装置で表示可能なオブジェクトシリアルナンバーの指定によって、データ取得履歴テーブル203から高速にキャプチャーデータの検索が可能になる。データ取得履歴テーブル203の使用に関してキャプチャーデータの消去を例にして説明した

が、もちろん再生の際も同様にキャプチャーデータの検索に用いることが可能である。また、画像、音声の区別なくキャプチャーした順にオブジェクトシリアルナンバーが付与され、しかも該オブジェクトシリアルナンバーが画像や音声のファイル名にも含まれるので、使用者はファイル名を見ただけでキャプチャーした順番を知ることができる。

【0043】本実施の形態によれば、第1に階層ヒープ構造を用いたマップファイルをによってキャプチャーデータおよびその属性情報を階層的に管理する手段を有することによって、検索等のために高速に取り出さなくてはならないデータに関しては浅い階層に配置し、取り出す頻度が低いデータに関しては深い階層に配置し、また、データのカテゴリ毎にヒープ単位でまとめて管理することで、不要なデータをまとめて読み飛ばす際のチェック回数が削減でき、可変長の属性情報を効率的に管理することができるようになり、検索速度を向上できるようになったとともに、少ないメインメモリを読み飛ばしによって効率的に使用することが可能になった。

【0044】高い読み込み速度が要求される属性情報は浅い階層に配置し、高い読み込み速度が要求されない属性情報は深い階層に配置することでマップファイル中の属性情報を用いた検索等の処理を高速化できる。さらに、高い読み込み速度が要求される属性情報はグループ化して浅い階層に配置し、高い読み込み速度が要求されない属性情報はグループ化して深い階層に配置することで、読み込みの必要の有りの判断を属性情報毎に行わずに済むため、マップファイル中の属性情報を用いた検索等の処理をさらに高速化できる。また、読み込む頻度が高い属性情報は浅い階層に配置し、読み込む頻度が低い属性情報は深い階層に配置することで、マップファイル中のデータ表示等の処理を高速化できる。さらに、読み込む頻度が高い属性情報はグループ化して浅い階層に配置し、読み込み頻度が低い属性情報はグループ化して深い階層に配置することで、読み込みの必要の有りの判断を属性情報毎に行わずに済むため、マップファイル中のデータ表示等の処理をさらに高速化できる。

【0045】また、マップファイルを階層ヒープ構造にすることでマップファイル中で管理する画像の単位を階層化する手段を有することで、連写やパノラマ撮影などによる複数画像を撮影単位毎で管理することを可能になり一覧表示のサムネール数を減らすことができ、複数の画像を撮影単位毎にまとめることによって、検索が容易になるとともに、複数の画像に対してまとめてディスクリプション等の付帯情報を付与することができるためデータ容量の節約が可能になった。

【0046】また、マップファイル中、小さなデータレコードは直接オフセットテーブル中に格納することでデータ格納スペースが節約できるのみならずデータレコードのアクセス速度が向上した。また、マップファイル中

の深い階層のヒープのデータレコードとして第1階層のヒープのデータレコードのインデックスを持つようなデータを格納方法を有することで1つのファイル中に階層的にデータを格納する際に階層構造に従って連続的にデータを配置する必要はないので階層構造の変更の際にファイル内でのデータの移動量を最小化することができた。また、データレコードを直接ヒープ中に持たず、外部ファイルに持つ、データレコード格納方法を有することで、大きなデータを外部ファイルに格納マップファイルの容量を小さくすることができる。

【0047】また、キャプチャーデータに対してその種別にかかわらずキャプチャーした順にシリアル番号をつけ、このシリアル番号によってデータを再生もしくは消去するデータを指定する手段を有することによって簡便な表示装置によって指定することが可能になった。

【0048】また、マップファイル中にキャプチャーデータに付与されたシリアル番号と、マップファイル中でのキャプチャーデータのインデックスの対応を記録するデータ取得履歴テーブルを有することで、種々のデータがマップファイル中に存在していても、キャプチャーデータに付与されたシリアル番号からマップファイル中で管理される該当キャプチャーデータに関するデータを高速に検索することが可能になった。

【0049】また、一つのマップファイルで管理可能なキャプチャーデータの最大数を限定し、キャプチャーデータが該最大数を越えた場合は新規なキャプチャーデータの格納単位を自動的に生成し、キャプチャーデータに付与するシリアル番号をリセットする手段を有することによって、表示装置の数字の表示桁数が限られている場合でも、多数のデータをキャプチャーし管理することが可能になった。

【0050】上記の説明では、映像データと音声データとの2種類のデータについて説明したが、他の2種類のデータについても本発明を適用することができる。

【0051】

【発明の効果】以上のように本発明によれば、処理したデータに、映像、音声に拘らず取得した順にシリアル番号を付加するように構成したので、シリアル番号の付加されたデータを記録媒体に記録した場合等において、再生又は消去するデータを簡単に検索することができる。

【0052】また、シリアル番号をリセットすることにより、シリアル番号を表示する場合、表示桁数が小さい簡単な構成の表示装置を用いることができる。また、処理したデータに、データ名あるいはファイル名を付加することにより、データ名やファイル名から容易にデータの検索を行うことができる。さらに、属性情報を階層ヒープ構造とすることにより、検索時に読み飛ばし等を行うことができ、検索を効率的に短時間で行うことができる。

【図1】デジタルカメラの記録媒体にキャプチャーデータを記録する際のディレクトリ及びファイルを示す構成図である。

【図2】デジタルカメラの構成を示すブロック図である。

【図3】デジタルカメラの背面を示す構成図である。

【図4】マップファイルに可変長データを格納する基本構造として用いるヒープ構造を示す構成図である。

【図5】レコード型フィールドの一般形を示す構成図である。

【図6】図4のヒープ構造を拡張した階層ヒープ構造を示す構成図である。

【図7】階層ヒープからのデータレコードのアクセス方法を示すフローチャートである。

【図8】マップファイルに格納されるデータ取得履歴テーブルの構成図である。

【図9】マップファイルに格納されるデータレコードを示す構成図である。

【図10】オフセットテーブルの構成図である。

【図11】マップファイルの中のイメージオブジェクトの構成図である。

【図12】サブオブジェクトのデータレコードの内部の構成図である。

【図13】イメージオブジェクトを階層ヒープ構造のマップファイルに格納する際の各プロパティの階層構造を示す構成図である。

【図14】Recモードの際のファイル管理の手順を示すフローチャートである。

【図15】図14のステップS2のカレントディレクトリ決定の処理の詳細なフローチャートである。

【図16】消去対象キャプチャーデータのオブジェクトシリアルナンバーを得た後のデータ消去の手順を示すフローチャートである。

【符号の説明】

107 CPU

108 記録媒体

109 操作部

122 シリアル番号表示部

125 消去ボタン

233 キャプチャーデータ保持ディレクトリ

239 マップファイル

240 画像データファイル

241 サムネール画像ファイル

242 音声データファイル

202 マップ管理オブジェクト

203 データ取得履歴テーブル

300 ヒープ構造

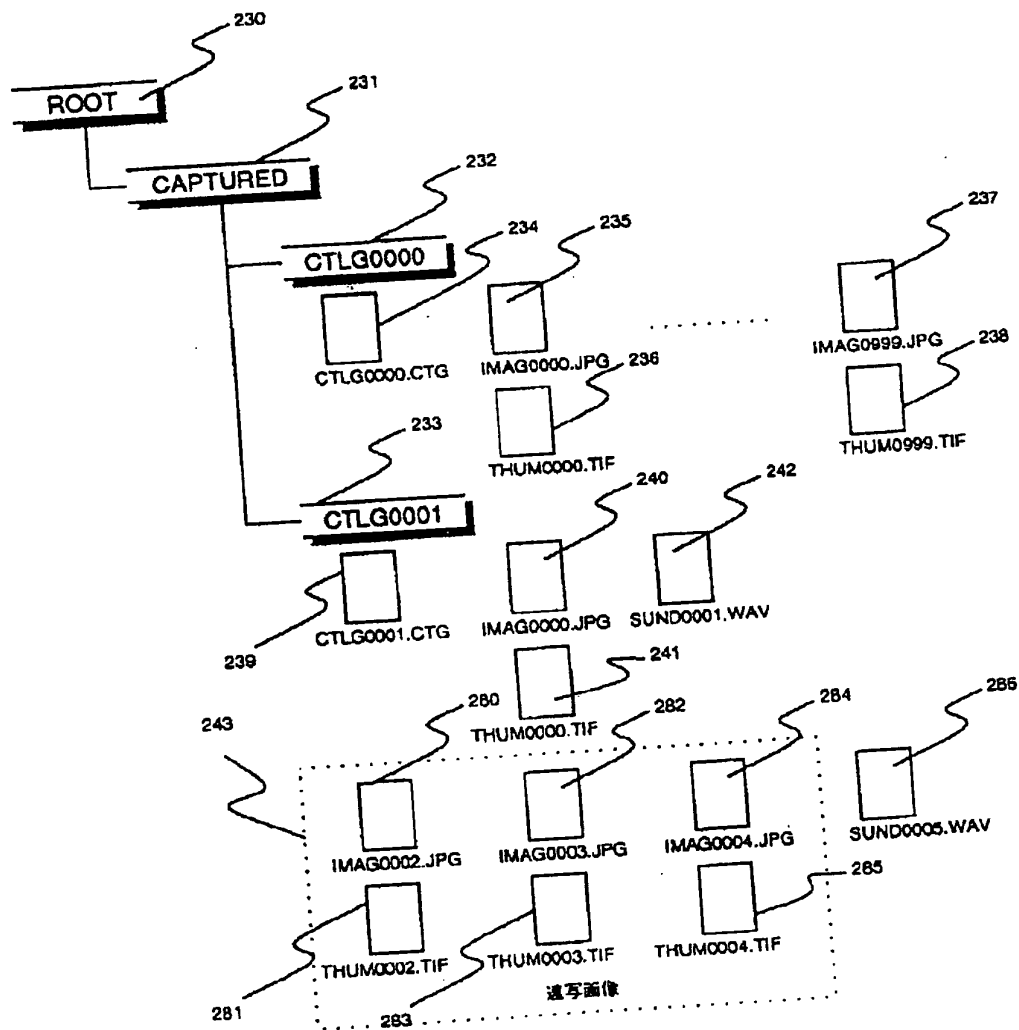
309 データレコード

305 データレコードのレコードオフセットフィールド

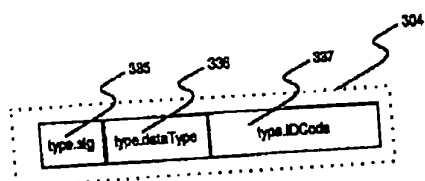
301 オフセットテーブル  
 350 オフセットテーブルオフセット

### 351 階層ヒープ構造

【図1】



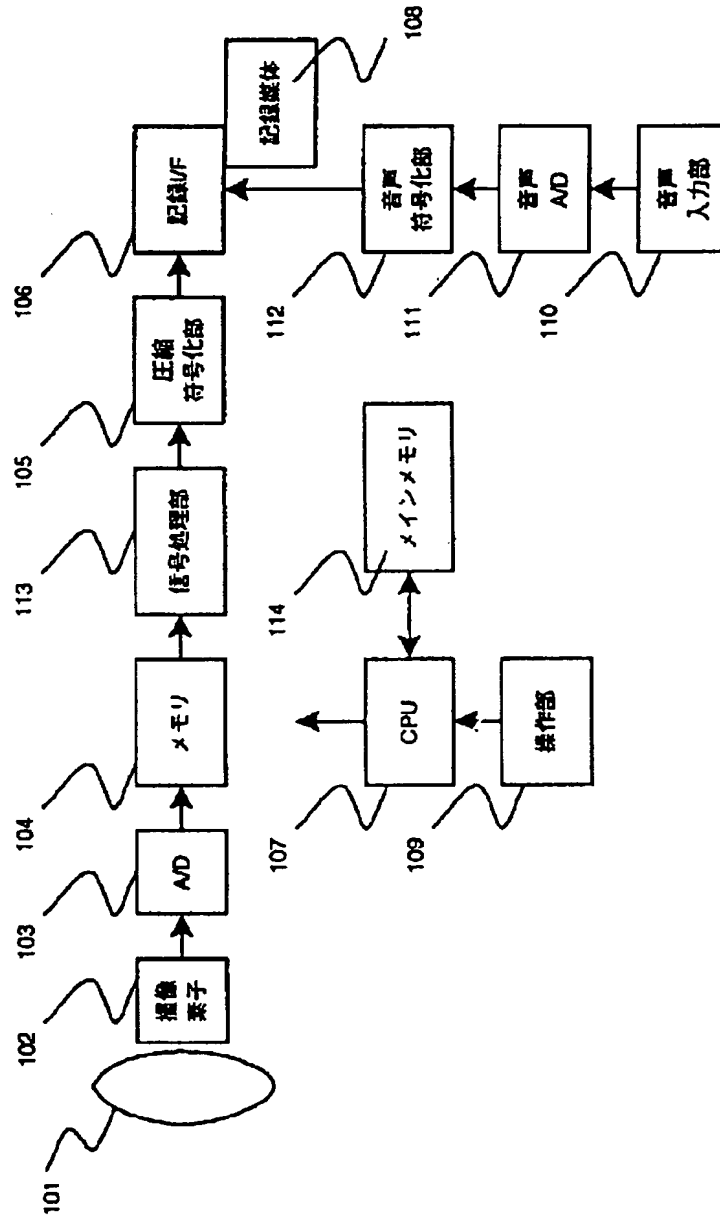
【図5】



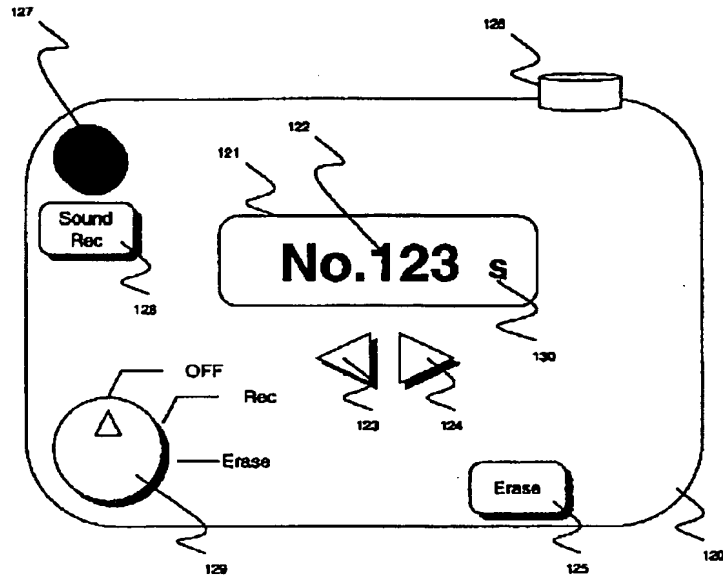
【図8】

Nobj	object index	subObject index
0	2	1
1	3	1
2	4	1
3	4	2

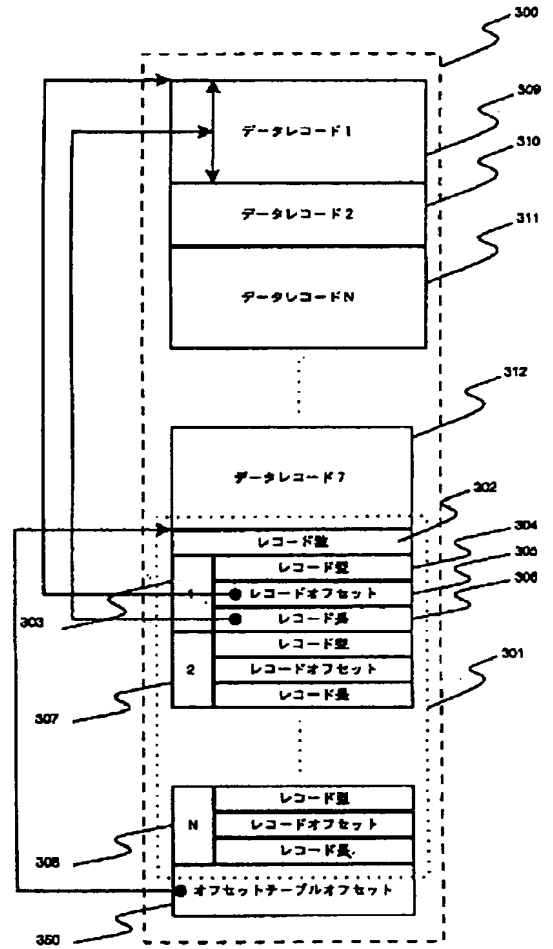
【図2】



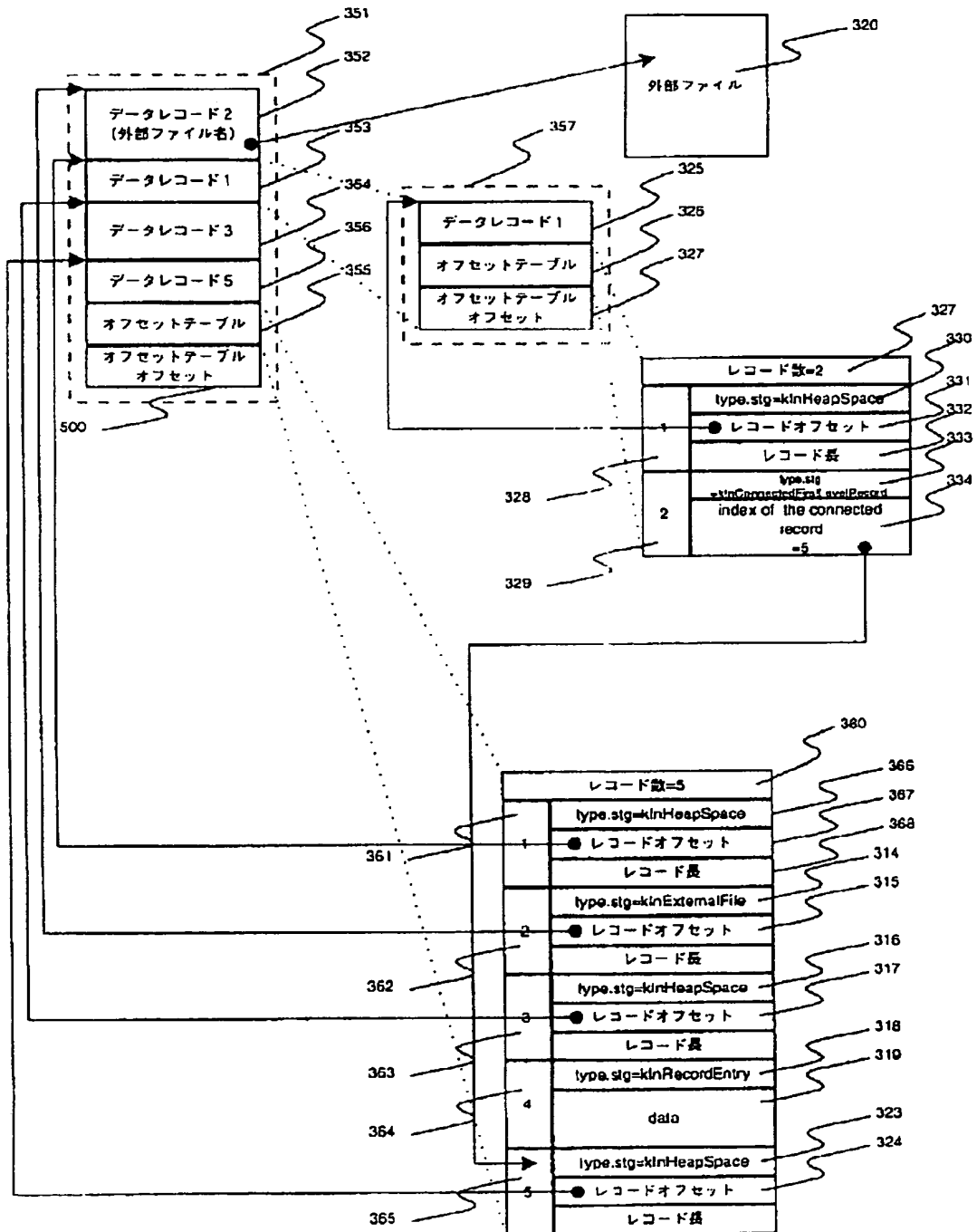
【図3】



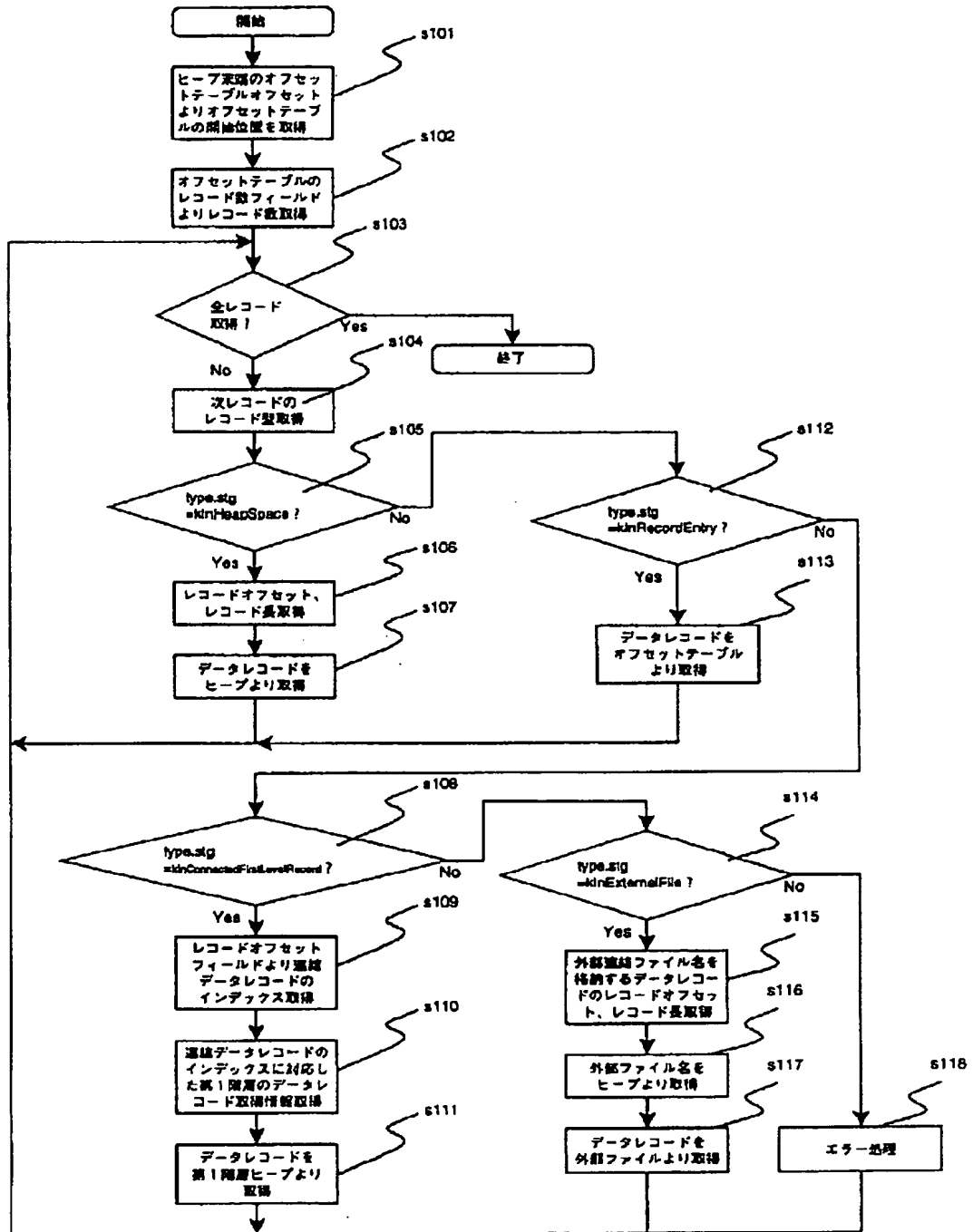
【図4】



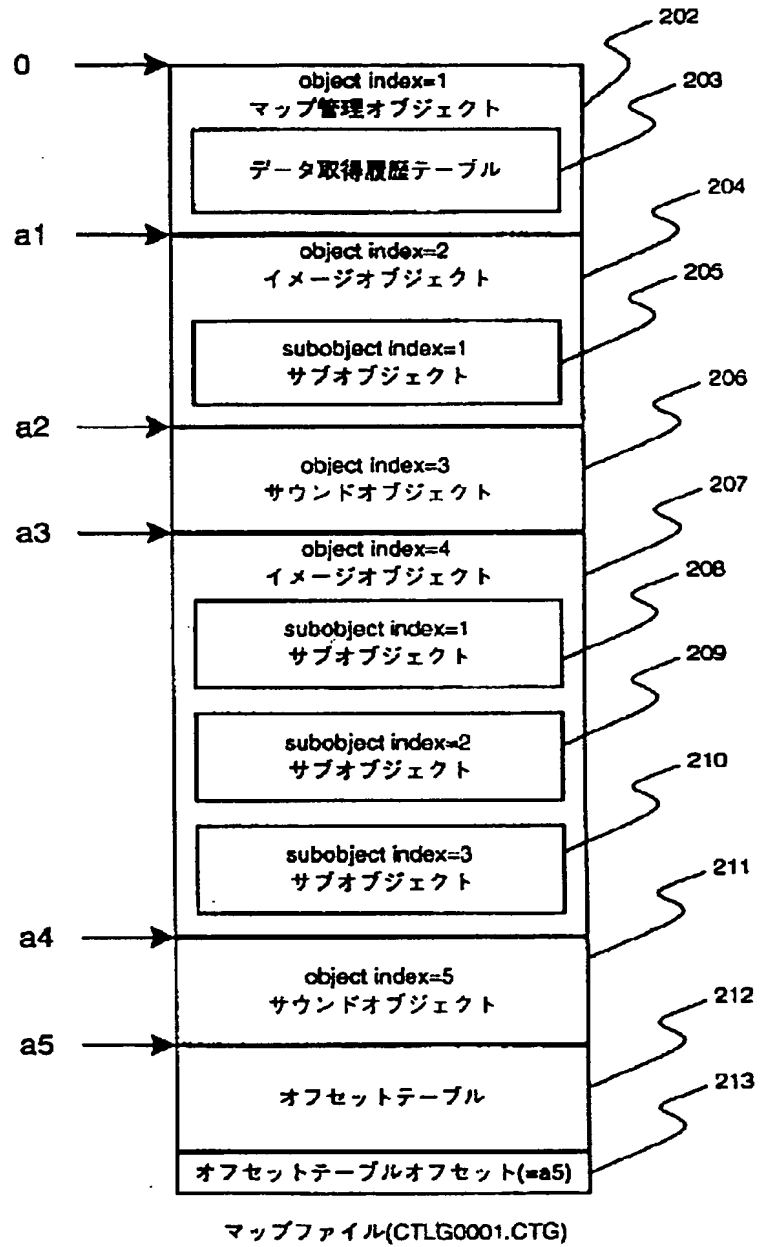
【図6】



【図 7】

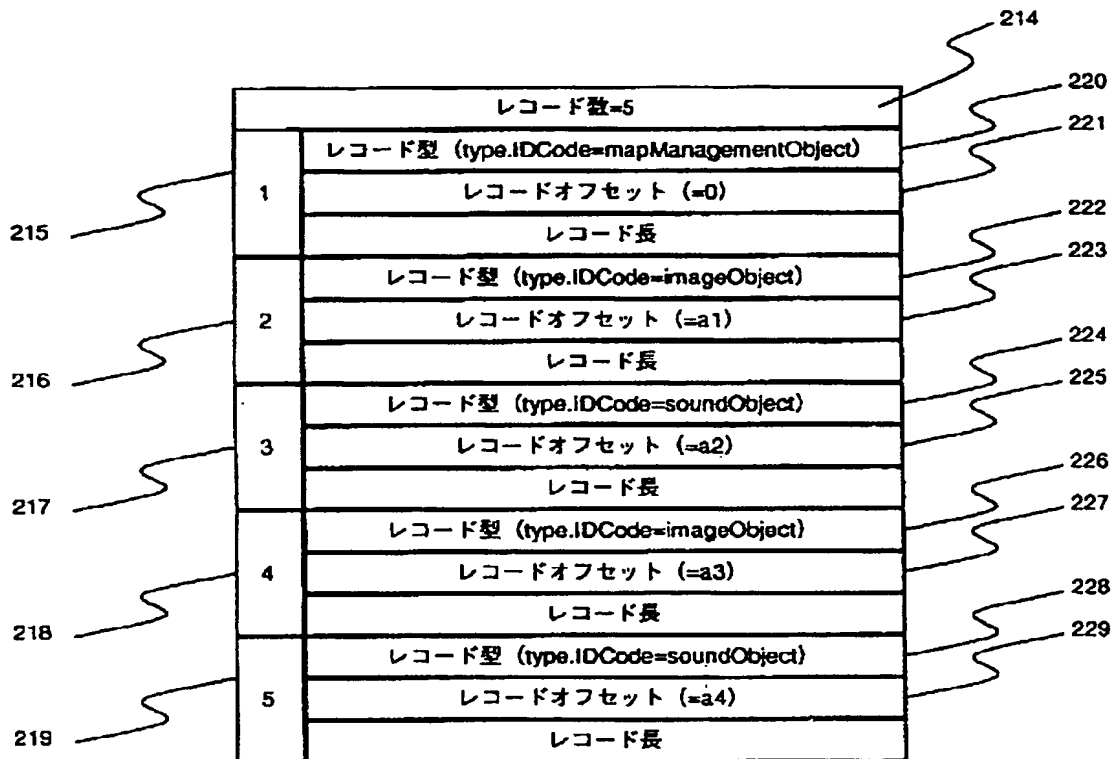


【図9】

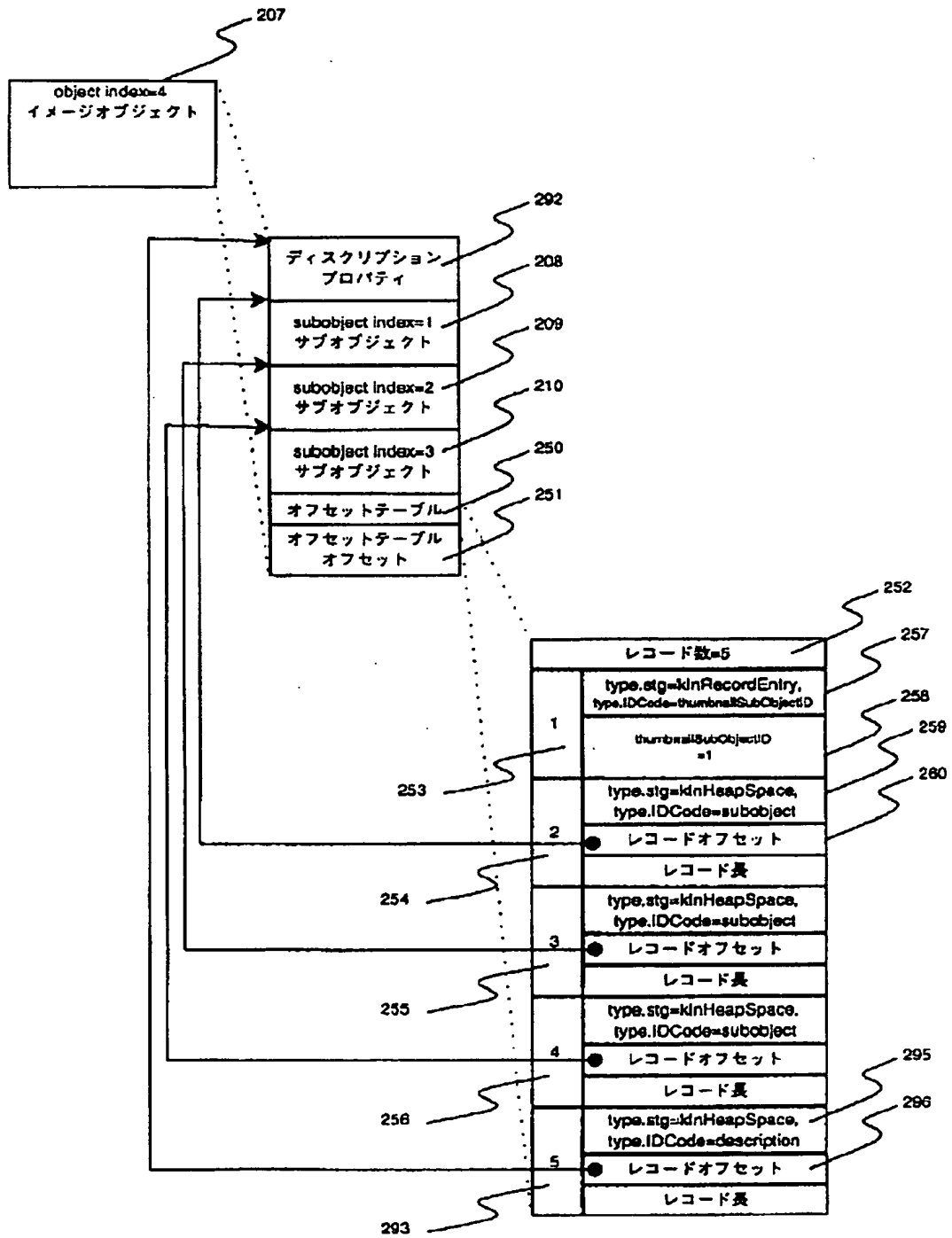




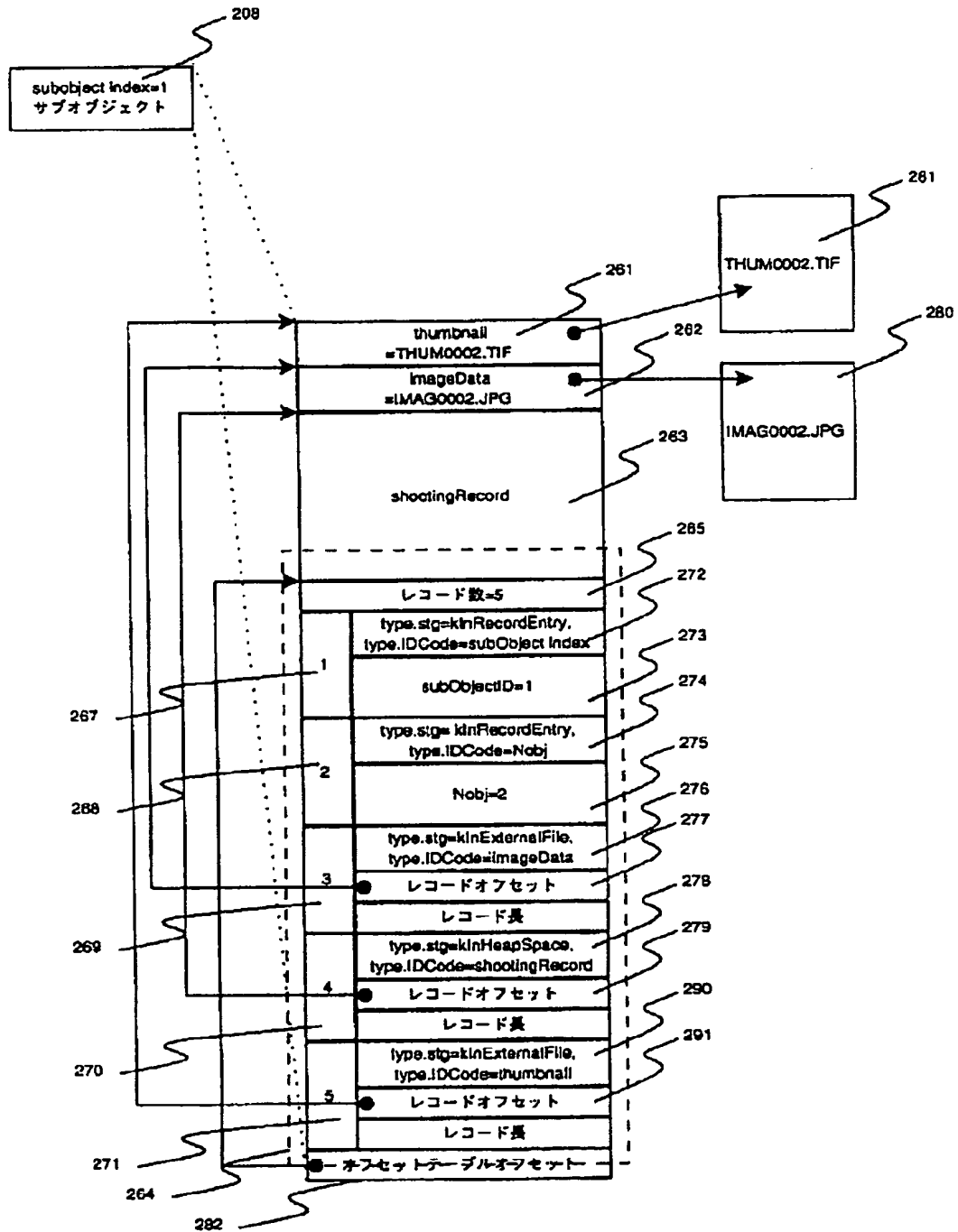
【図10】



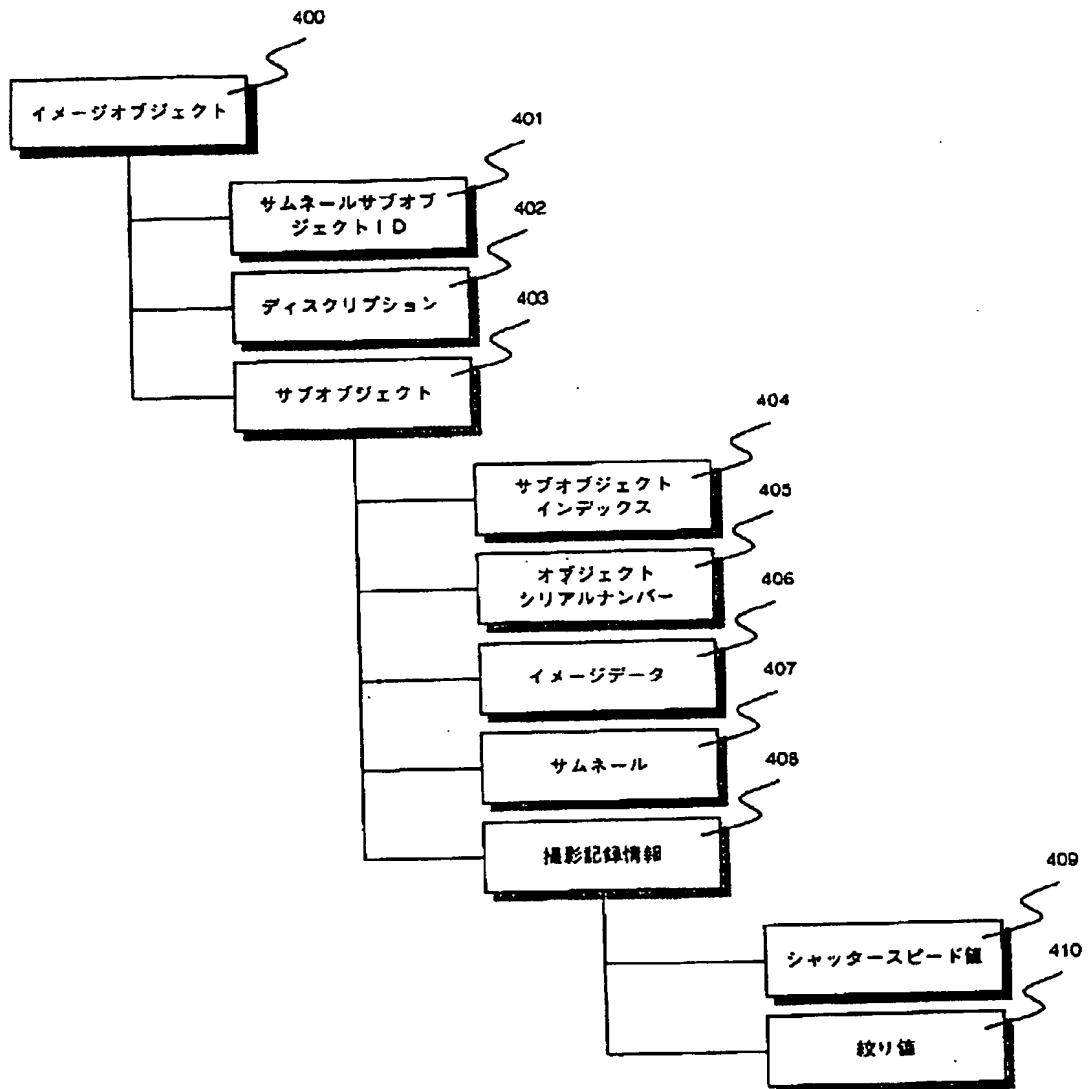
【図 11】



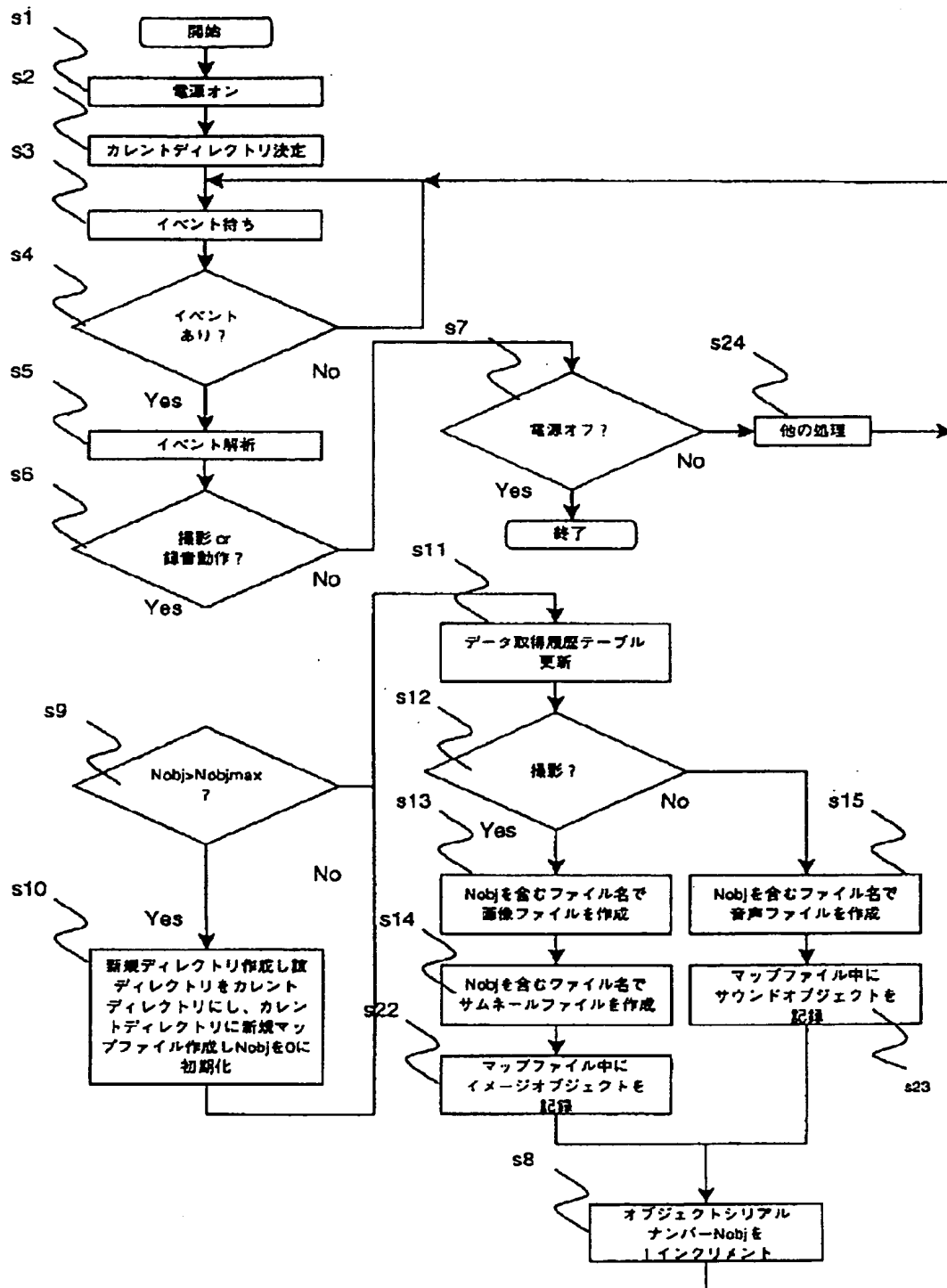
【図12】



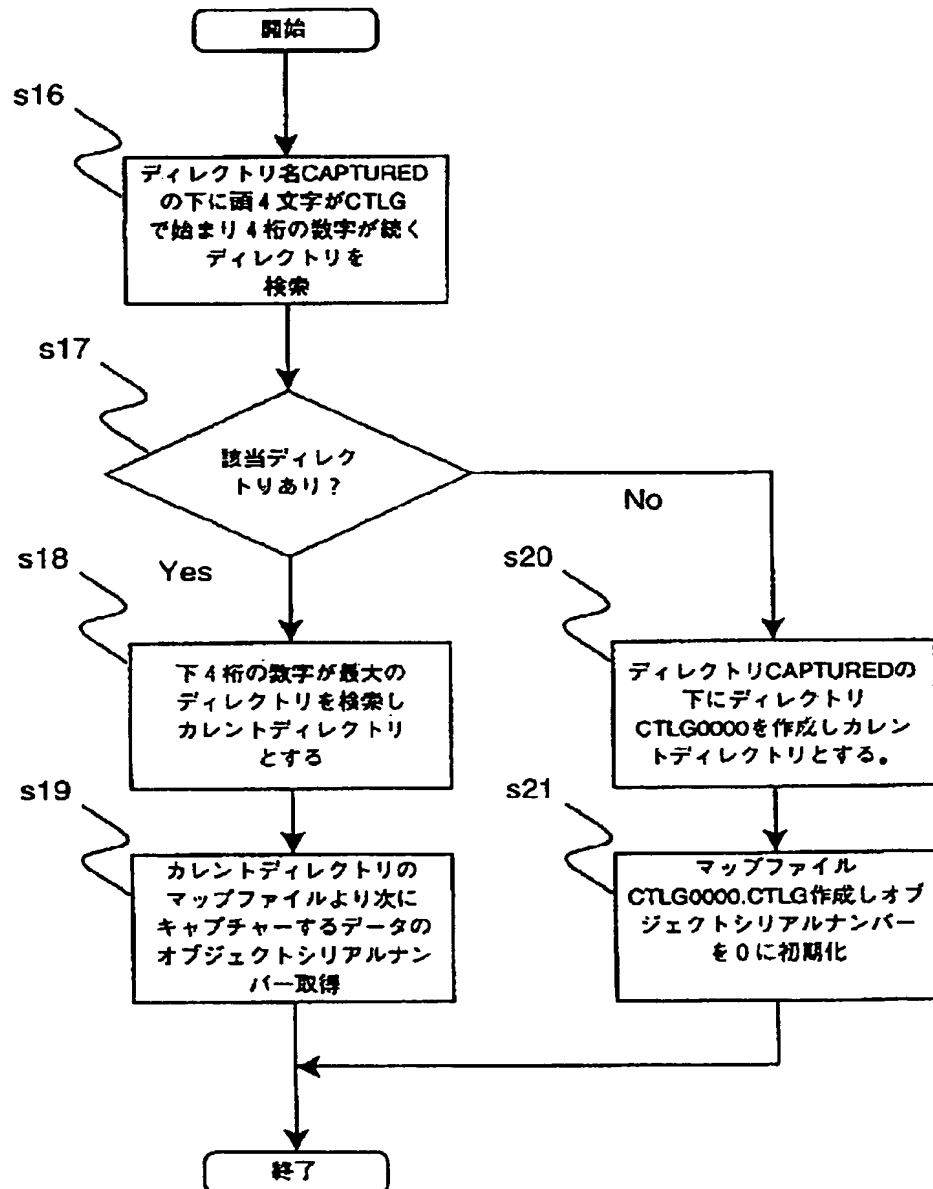
【図13】



【図14】



【図15】



【図16】

